WHAT IS CLAIMED IS:

1. In a database system, a method for constructing an optimal query execution plan for executing a query, the method comprising:

receiving a query specifying at least one join condition between two or more database tables;

identifying each query block within said query, each query block comprising an atomic portion of said query;

creating subplans for each query block based on grouping portions of each query block;

determining at least one favorable access plan for each subplan of each query block, said at least one favorable access plan determined based at least in part on estimated execution costs;

generating an optimal access plan for each query block based upon said at least one favorable access plan determined for each subplan; and

constructing an optimal query execution plan based upon said optimal access plan generated for each query block.

2. The method of claim 1, wherein said step of identifying each query block within said query includes building a query optimization graph for each query block.

3. The method of claim 1, further comprising:

generating a query optimization graph for each query block.

4. The method of claim 3, wherein said step of generating a query optimization graph includes generating subplans for each query block.

5. The method of claim 3, wherein said step of generating a query optimization graph includes generating plan nodes for each subplan, said plan nodes for joining tables and subplans.

6. The method of claim 5, wherein said step of generating a query optimization graph includes generating an array of access methods for each plan node.

7. The method of claim 5, wherein said step of generating a query optimization graph includes generating an array of join methods for each plan node.

8. The method of claim 1, wherein a subplan represents a table expression of a query block.

9. The method of claim 1, wherein said step of determining at least one favorable access plan for each subplan includes generating at least one access plan for each different set of outer references.

10. The method of claim 1, wherein the step of determining at least one favorable access plan for each subplan includes the substeps of:

placing a candidate plan segment in the next position in a current access plan being generated, said candidate plan segment representing a particular plan node, access method and join method valid at said next position;

evaluating the current access plan including said candidate plan segment; and

if the current access plan is less favorable than a favorable access plan previously identified, replacing said candidate plan segment with another available candidate plan segment and repeating said evaluating substep.

11. The method of claim 10, wherein a plan node comprises a quantifier object.

12. The method of claim 11, wherein said quantifier object represents a base table.

13. The method of claim 11, wherein said quantifier object represents a derived table.

14. The method of claim 10, wherein a plan node comprises a subplan object.

15. The method of claim 14, wherein said subplan object represents a join tree of a table expression of a query block.

16. The method of claim 10, further comprising:

if the current access plan is more favorable than a favorable access plan previously identified, determining whether the current access plan comprises a complete plan;

if the current access plan is determined not to comprise a complete plan, retaining the current access plan and repeating the above substeps for placing a candidate plan segment in the next position of the current access plan; and

otherwise, if the current access plan is determined to be a complete plan, retaining the current access plan as a favorable access plan and repeating the above substeps to consider other available alternatives to the current access plan while alternatives are available.

17. The method of claim 10, further comprising:

generating a property vector for said current access plan including said candidate plan segment.

18. The method of claim 17, wherein said property vector includes estimated execution costs for said current access plan.

19. The method of claim 17, wherein said property vector includes pipeline characteristics for said current access plan.

20. The method of claim 17, wherein said property vector includes order properties for said current access plan.

21. The method of claim 1, wherein said step of generating at least one favorable access plan for each subplan of each query block includes using a left-deep join enumeration strategy.

22. The method of claim 1, wherein said step of generating at least one favorable access plan for each subplan of each query block includes starting with the innermost nested subplans.

23. The method of claim 1, wherein said step of generating at least one favorable access plan for each subplan includes evaluating execution costs of partial access plans and pruning partial access plans less favorable than previously generated complete access plans.

24. A computer-readable medium having computer-executable instructions for performing the method of claim 1.

25. A downloadable set of computer-executable instructions for performing the method of claim 1.

26. In a database system, a method for generating a bushy trees during optimization of a database query, the method comprising:

receiving a database query specifying at least one join condition between two or more database tables;

identifying each query block within said query, said query block comprising an atomic block of said query;

building a query optimization graph for each query block, said query optimization graph including plan nodes representing subplans and quantifiers of each query block;

constructing a join tree for each subplan based upon selecting access methods, join methods, and join order for plan nodes of said query optimization graph having favorable execution costs;

constructing an optimal access plan for each query block based upon said join tree constructed for each subplan; and

generating a bushy execution tree based upon the optimal access plan determined for each query block.

27. The method of claim 26, wherein a query block comprises a selected one of a main block of a Structured Query Language (SQL) statement, a main block of a derived table, a main block of a view, a main block of a subquery used in a SQL statement, a derived table, and a view.

28. The method of claim 26, wherein said step of building said query optimization graph includes modeling left outer joins, right outer joins, and full outer joins as subplans which correspond to null-supplying sides of an outer join.

29. The method of claim 26, wherein said step of generating a query optimization graph includes generating an array of access methods for each plan node.

30. The method of claim 26, wherein said step of generating a query optimization graph includes generating an array of join methods for each plan node.

31. The method of claim 26, wherein said step of constructing bushy join trees for each subplan includes using a left-deep enumeration strategy.

32. The method of claim 26, wherein said step of constructing a join tree includes evaluating execution costs of a candidate plan segment to be added to said join tree being constructed.

33. The method of claim 32, wherein said candidate plan segment comprises a selected plan node together with an access method and a join method.

34. The method of claim 26, wherein said bushy tree comprises a processing tree having composite relations for left and right children of join nodes.

35. The method of claim 26, wherein said step of constructing an optimal bushy access plan includes evaluating execution costs of partial access plans enabling earlier pruning of unfavorable access plans.

36. In a database system, a method for optimizing execution of a query, the method comprising:

receiving a query specifying selection of data from a plurality of database tables;

enumerating candidate plan segments for inclusion in an access plan for selecting data specified by the query, said candidate plan segments representing alternative strategies for joining relations and selecting data;

for each query block comprising an atomic portion of said query, determining an optimal access plan by performing the substeps of:

placing a candidate plan segment in a partial access plan being generated for said query block;

evaluating said partial access plan including said candidate plan segment;

if said partial access plan is less favorable than a complete access plan previously identified for said query block, pruning said candidate plan segment;

otherwise, adding an additional candidate plan segment to said partial access plan and repeating the above steps until a complete access plan for said query block is generated;

retaining a complete access plan if it is more favorable than other complete access plans previously generated for the query block; and

otherwise pruning a complete access plan which is less favorable than other complete access plans; and

generating a query execution plan based upon the optimal access plan determined for each query block.

37. The method of claim 36, wherein each said candidate plan segment comprises a plan node, an access method and a join method.

38. The method of claim 37, wherein a plan node comprises a subplan object.

39. The method of claim 37, wherein a plan node comprises a quantifier object.

40. The method of claim 36, where said substep of evaluating said partial access plan includes comparing estimated execution costs of said partial access plan to estimated execution costs of a complete access plan previously generated for said query block.

41. The method of claim 36, where said substep of evaluating said partial access plan before a complete access plan is generated enables early pruning of less favorable access plans.

42. The method of claim 36, wherein said step of generating an optimal access plan for each query block includes generating an optimal access plan for each subplan of each query block.

43. The method of claim 42, wherein said step of generating an optimal access plan for each subplan of each query block includes starting with the innermost nested subplans.

44. The method of claim 36, wherein said step of generating an optimal access plan for each query block includes using a left-deep join enumeration strategy.

45. The method of claim 36, further comprising:

generating a property vector for each said partial access plan.

46. The method of claim 45, wherein said property vector includes estimated execution costs for a candidate plan segment.

47. The method of claim 45, wherein said evaluating substep includes comparing said property vector of said partial access plan to a property vector of a complete access plan previously generated for said query block.

48. The method of claim 45, further comprising:

retaining a property vector of a complete plan generated for a query block if said complete plan is more favorable than any other complete access plan previously generated for the query block.

49. The method of claim 36, wherein said substep of pruning said candidate plan segment includes replacing said candidate plan segment with another available candidate plan segment and repeating said evaluating substep.